17 MARS 2022 LSI1 S2

RAPPORT SUR LA MISE EN PLACE D'UN CRUD ETUDIANT

----ATELIER 1-----

CRUD MVC PHP FRAMEWORK

- REALISE PAR : AMAL TOUHAMI,
  étudiante 1 ere année en cycle Ingenieure LSI1 FST Tanger
- ENCADRE par : Pr. LOTFI EL AACHAK,
  notre professeur et encadrant au module Developpement Web& Frameworks

Objectifs

L'objectif principal de cet atelier est la mise en place d'un CRUD contenant les opérations création, mise a jour, suppression et sélection a travers le framework développé from scratch.

Nous mettrons en œuvre ainsi dans cet atelier les principes du Modele MVC.

Travail à faire :

1. Créer la BDD ainsi une table étudiant « id_etudiant , nom , prenom , age, cne

2. Créer le model en développent les classes : Etduaiant EtudiantTrasaction.

3. Developper le contrôleur étudiant , ainsi modifier le fichier Route.php en ajoutant les actions adéquats aux opérations CRUD.

4. Développer le deux interface graphique web , le formulaire de création et la liste des étudiants avec les actions modifier et supprimer.

PLAN:

Vous trouverez les fichiers source associés à l'adresse :
https://github.com/myyla/ATELIER-1--CRUD-ETUDIANT

INTRODUCTION:

De manière générale, tout logiciel doit gérer plusieurs problématiques :

- Interactions avec l'extérieur, en particulier l'utilisateur : saisie et contrôle de données, affichage. C'est la problématique de présentation .
- Opérations sur les données (calculs) en rapport avec les règles métier . C'est la problématique des traitements .
- Accès et stockage des informations qu'il manipule, notamment entre deux utilisations. C'est la problématique des données.

Quand une page web mélange code de présentation (les balises HTML) et accès aux données (requêtes SQL). Ceci sera contraire au principe de responsabilité unique. Ce principe de conception logicielle est le suivant : afin de clarifier l'architecture et de faciliter les évolutions, une application bien conçue doit être décomposée en sous-parties, chacune ayant un rôle et une responsabilité particuliers. L'architecture actuelle montre ses limites dès que le contexte se complexifie. Le volume de code des pages PHP explose et la maintenabilité devient délicate. Il faut faire mieux.

----------> D ou le passage a l'Architecture MVC.

Présentation du modèle MVC

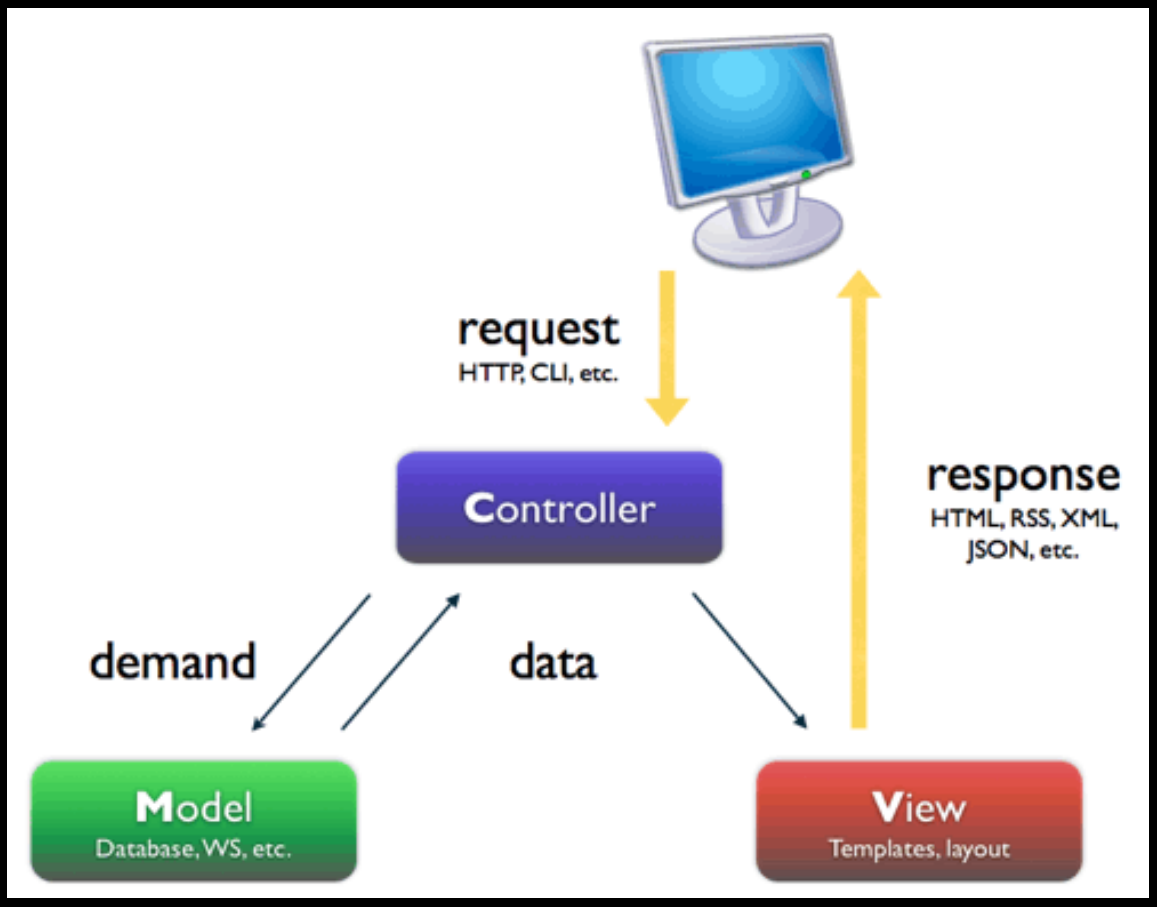
Le modèle MVC décrit une manière d'architecturer une application informatique en la décomposant en trois sous-parties :

- la partie Modèle ;
- la partie Vue ;
- la partie Contrôleur.

Ce modèle de conception (« *design pattern* ») a été imaginé à la fin des années 1970 pour le langage Smalltalk afin de bien séparer le code de l'interface graphique de la logique applicative. Il est utilisé dans de très nombreux langages : bibliothèques Swing et Model 2 (JSP) de Java, *frameworks* PHP, ASP.NET MVC, etc.

Roles des composantes MVC

La partie Modèle d'une architecture MVC encapsule la logique métier (« business logic ») ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).
La partie Vue s'occupe des interactions avec l'utilisateur : présentation, saisie et validation des données.
La partie Contrôleur gère la dynamique de l'application. Elle fait le lien entre l'utilisateur et le reste de l'application.



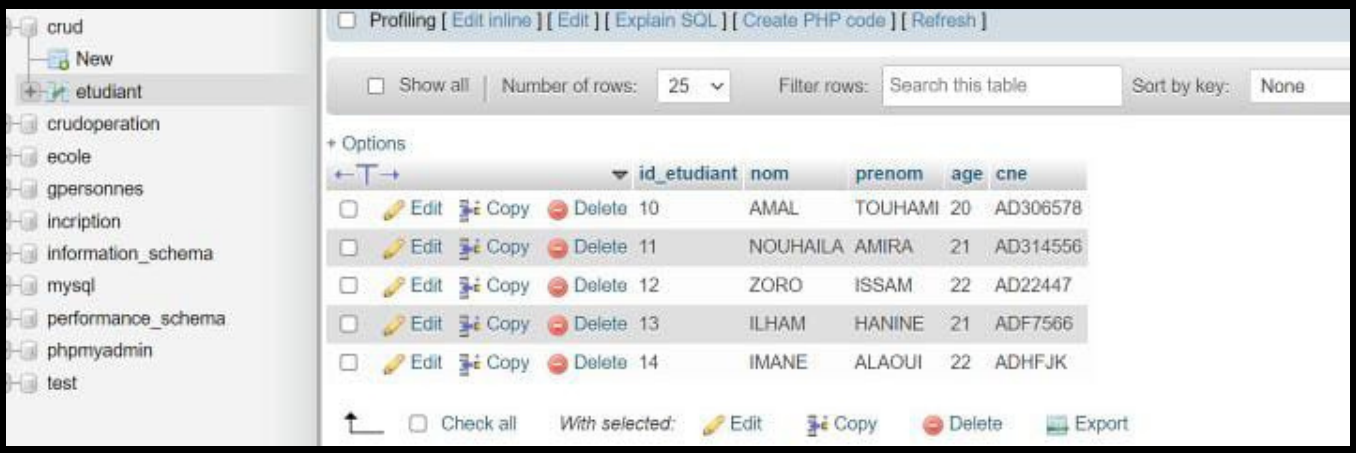
- Le diagramme ci-dessus résume les relations entre les composants d'une architecture MVC.

1. La demande de l'utilisateur (exemple : requête HTTP) est reçue et interprétée par le Contrôleur.
2. Celui-ci utilise les services du Modèle afin de préparer les données à afficher.
3. Ensuite, le Contrôleur fournit ces données à la Vue, qui les présente à l'utilisateur (par exemple sous la forme d'une page HTML).

- Une application construite sur le principe du MVC se compose toujours de trois parties distinctes. Cependant, il est fréquent que chaque partie soit elle-même décomposée en plusieurs éléments. On peut ainsi trouver plusieurs modèles, plusieurs vues ou plusieurs contrôleurs à l'intérieur d'une application MVC.

CREATION DE LA BASE DE DONNEES

- La base de données utilisée portant le nom "crud" est très simple. Elle se compose d une seule table, la table etudiant « id_etudiant , nom , prenom , age, cne ».



CREATION DU MODEL : CLASSE ETUDIANT

- Modele.php représente la partie Modèle (accès aux données)
- La classe etudiant represente la table Etudiant dans la base de donnees.

```
<?php

class Etudiant{

 private $id_Etudiant;
 private $nom;
 private $prenom;
 private $age;
 private $cne;

 public function __construct ( $nom=NULL, $prenom=NULL, $age=NULL , $cne=NULL){

  $this->nom = $nom;
  $this->prenom = $prenom;
  $this->age = $age;
  $this->cne = $cne;
 }


 public function getId() {
  return $this->id_Etudiant;
 }

 public function getNom() {
  return $this->nom;
 }

 public function getPrenom() {
  return $this->prenom;
 }

 public function getAge() {
  return $this->age;
 }

 public function getCne() {
  return $this->cne;
 }
}
```

CREATION DU MODEL:

CLASSE ETUDIANTRANSACTION

```php
<?php
require_once "Etudiant.php";

class EtudiantTransaction{

    private $db;    // Instance de PDO

    public function __construct(){
        try
        {
          $this->db= new PDO('mysql:host=localhost;dbname=crud;charset=utf8', 'root', '');
        }
        catch(Exception $e) {
            die('Erreur : '.$e->getMessage());
        }
    }

    // POUR L'Ajout d'un Etudiant
    public function save($etudiant){

        $q= $this->db->prepare('INSERT INTO etudiant(nom, prenom,age,cne ) VALUES(:nom, :prenom, :age,:cne)');

        $q->bindValue(':nom', $etudiant->getNom());
        $q->bindValue(':prenom', $etudiant->getPrenom() );
        $q->bindValue(':age', $etudiant->getAge(), PDO::PARAM_INT);
        $q->bindValue(':cne', $etudiant->getCne());
        $q->execute();

        header('Location: http://localhost/CRUD-ETUDIANT/index.php?action=etudiant');
    }

    //POUR LA SUPPRESSION
    public function delete($cne){
      $q=$this->db->prepare("DELETE FROM etudiant WHERE cne=:cne");
      $q->execute(array('cne'=>$cne));

      header('Location: http://localhost/CRUD-ETUDIANT/index.php?action=etudiant');
    }

     //POUR LA MODIFICATION
     public function update($cne){
        $q=$this->db->prepare("UPDATE etudiant SET nom=:nom,prenom=:prenom,age=:age,cne=:cne WHERE cne=:old_cne");

        $q->execute(array( 'nom' =>$_POST['nom'],
        'prenom' => $_POST['prenom'],
        'age' => $_POST['age'],
        'cne'=>$_POST['cne'],
        'old_cne'=>$cne
        ));
    header('Location: http://localhost/CRUD-ETUDIANT/index.php?action=etudiant');
    }

    //POUR SELECTIONNER UNE SEULE LIGNE
    public function getrow($cne){
    $q=$this->db->prepare("SELECT * FROM etudiant WHERE cne=:cne");
    $q->execute(array('cne'=>$cne));
    $row=$q->fetch(PDO::FETCH_ASSOC);
    return $row;
    }

    //POUR SELECTIONNER TOUS
    public function getAll(){
    $q  = $this->db->query('SELECT * FROM etudiant ORDER BY nom');
    $etudiants =  array();

    while ($donnees = $q->fetch(PDO::FETCH_ASSOC)) {
        $p= new Etudiant($donnees['nom'] , $donnees['prenom'] , $donnees['age']  , $donnees['cne']);
        array_push($etudiants,$p);
    }
    return $etudiants;
     }

     //---------------------------------
    public function setDb(PDO $db) {
        $this->db = $db;

    }
}
```

DEVELOPPEMENT DU CONTROLEUR: ControllerEtudiant

```
<?php
require_once 'Model/Etudiant.php';
require_once 'Model/EtudiantTransaction.php';
require_once 'View/view.php';

class ControllerEtudiant{
   private $etudiant;
   private $etudiantTransaction;

public function __construct(){
$this->etudiant = new Etudiant();
$this->etudiantTransaction= new EtudiantTransaction();
}


public function getEtus() {
   $etudiants = $this->etudiantTransaction->getAll();
   $view = new view("Etudiant");
   $view->generer(array('etudiants' => $etudiants));
}


public function addE() {


   $view = new view("AddEtudiant");
   $view->generer(array());
}


public function saveE() {
   $etudiant=$this->etudiant =new
Etudiant($_POST['nom'],$_POST['prenom'],$_POST['age'],$_POST['cne']) ;
   $this->etudiantTransaction->save($this->etudiant);
   getEtus();
}


public function deleteE(){
   $this->etudiantTransaction->delete($_POST['cne_delete']);
}


public function updateE($cne){
   $this->etudiantTransaction->update($cne);
}
public function getOne($cne) {
  $row = $this->etudiantTransaction->getrow($cne);
  $view = new view("Update");
  $view->generer(array('row' => $row));
}
}
?>
```

Route.php


```
<?php
 require_once 'Controller/ControllerEtudiant.php';
 require_once 'View/view.php';

 class  Route{

 private $ctrlEtudiant;

 public function __construct() {
   $this->ctrlEtudiant = new ControllerEtudiant();
}


public function routeRequete() {

try {
      if (isset($_GET['action']))
      {

            if ($_GET['action'] == 'etudiant') {
                  $this->ctrlEtudiant->getEtus();
            }
            else if($_GET['action'] == 'add'){

            $this->ctrlEtudiant->addE();

           } else if($_GET['action'] == 'save'){
           //var_dump($_POST);
           //exit;
           $this->ctrlEtudiant->saveE();

           }else if($_GET['action'] == 'delete'){
           //var_dump($_POST);
           //exit;
           $this->ctrlEtudiant->deleteE();
           }
           else if($_GET['action'] == 'update'){
           //var_dump($_POST);
           //exit;
           $this->ctrlEtudiant->updateE($_POST['old_cne']);
           }
           else if($_GET['action'] == 'getrow'){
           //var_dump($_POST);
           //exit;
            $this->ctrlEtudiant->getOne($_POST['cne_update']);
      }
      else{
         throw new Exception("Action non valide");
      }
   }
}
catch (Exception $e) {

      $this->erreur($e->getMessage());
}
 }
  // Affiche une erreur
 private function erreur($msgErreur) {

 $view = new view("Error");
 $view->generer(array('msgErreur' => $msgErreur));

 }

}
```

DEVELOPPEMENT DES INTERFACES GRAPHIQUES WEB

Gabarit.php

- Les gabarits sont là pour vous faire gagner du temps : vous pouvez paramétrer les différents éléments de votre classe , puis créer vos nouvelles classes à partir de ce gabarit, en un clic.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<link rel="stylesheet" href="Contenu/style.css" /> <title><?= $titre ?></title>
  </head>
  <body>


  <div class="container">
  <div id="global">
      <!-- <header>
<a href="index.php"><h1 id="titreBlog">page
Personne</h1></a> </header> -->


<h1 style="text-align: center; margin-top: 30px;" >Liste des Etudiants  </h1>
<hr>
      <div id="contenu">
        <?= $contenu ?>

  </div>
      <!-- #contenu -->
  </div>

      </div> <!-- #global -->
  </body>
</html>
```

DEVELOPPEMENT DES INTERFACES GRAPHIQUES WEB

view.php

```
<?php
class view {

   private $fichier;
   private $titre;

   public function __construct($action) {
      // Détermination du nom du fichier vue à partir de
       $this->fichier = "View/view" . $action . ".php";
   }

   public function generer($donnees) {
      $contenu = $this->genererFichier($this->fichier, $donnees);
      $view = $this->genererFichier('View/Gabarit.php',array('titre' => $this->titre, 'contenu' => $contenu));
      echo $view;
   }

   private function genererFichier($fichier, $donnees) {
      if (file_exists($fichier)) {
         extract($donnees);
         ob_start();
         require $fichier;

         return ob_get_clean();

      }
      else {
         throw new Exception("Fichier '$fichier' introuvable");
      }
   }

}

?>
```

DEVELOPPEMENT DES INTERFACES GRAPHIQUES WEB

viewAddEtudiant.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
integrity="sha384-
B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l"
crossorigin="anonymous">
    <title>Ajouter un etudiant </title>
</head>
<body>

<div class="container mt-5">
    <form method="POST" action="http://localhost/CRUD-ETUDIANT/index.php?
action=save">
    <div class="form-group">
        <label for="nom">Nom:</label>
        <input type="text" class="form-control" id="nom"  name="nom">
    </div>
    <div class="form-group">
        <label for="prenom">Prenom:</label>
        <input type="text" class="form-control" id="prenom" name="prenom">
    </div>
    <div class="form-check">
        <label for="age">Age:</label>
        <input type="number" class="form-control" id="age" name="age">
    </div>
    <div class="form-check">
        <label for="cne">CNE:</label>
        <input type="text" class="form-control" id="cne" name="cne">
    </div>
    <button type="submit" class="btn btn-primary">
      Ajouter
    </button>
    </form>
</div>


</body>
</html>
```

DEVELOPPEMENT DES INTERFACES GRAPHIQUES WEB

viewDeleteEtudiant.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <?php

    var_dump($_POST);

    ?>
</body>
</html>
```

DEVELOPPEMENT DES INTERFACES GRAPHIQUES WEB

viewEtudiant.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-
B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l" crossorigin="anonymous">
    <title>Document</title>
</head>
<body>
<div class="container">
<a href="http://localhost/CRUD-ETUDIANT/index.php?action=add" class="btn btn-success my-3">Ajouter un Etudiant</a>

<table class="table">
  <thead class="thead-light">
    <tr>
      <th scope="col">Nom</th>
      <th scope="col">Prenom</th>
      <th scope="col">Age</th>
      <th scope="col">CNE</th>
      <th scope="col">Operations</th>
    </tr>
  </thead>
  <tbody>
  <?php foreach ($etudiants as $etudiant): ?>

    <tr>


      <td><?php echo $etudiant->getNom();?></td>
      <td><?php echo $etudiant->getPrenom();?></td>
      <td><?php echo $etudiant->getAge();?></td>
      <td><?php echo $etudiant->getCne();?></td>

      <td>
 <?php $tab=array("nom"=>$etudiant->getNom(),"prenom"=>$etudiant->getPrenom(),"age"=>$etudiant->getAge(),"cne"=>$etudiant-
>getCne()) ?>


 <form action="http://localhost/CRUD-ETUDIANT/index.php?action=getrow" method="POST" >
 <button class="btn btn-primary mb-1" value="<?php echo $etudiant->getCne();?>" name="cne_update">
 Modifier
 </button>
 </form>
 <form action="http://localhost/CRUD-ETUDIANT/index.php?action=delete" method="POST" >
 <button class="btn btn-danger px-2" value="<?php echo $etudiant->getCne();?>" name="cne_delete">
 Delete
 </button>
 </form>
 </td>
 </tr>
 <?php endforeach; ?>

 </tbody>
</table>
</div>

</body>
</html>
```

DEVELOPPEMENT DES INTERFACES GRAPHIQUES WEB

ViewUpdate.php

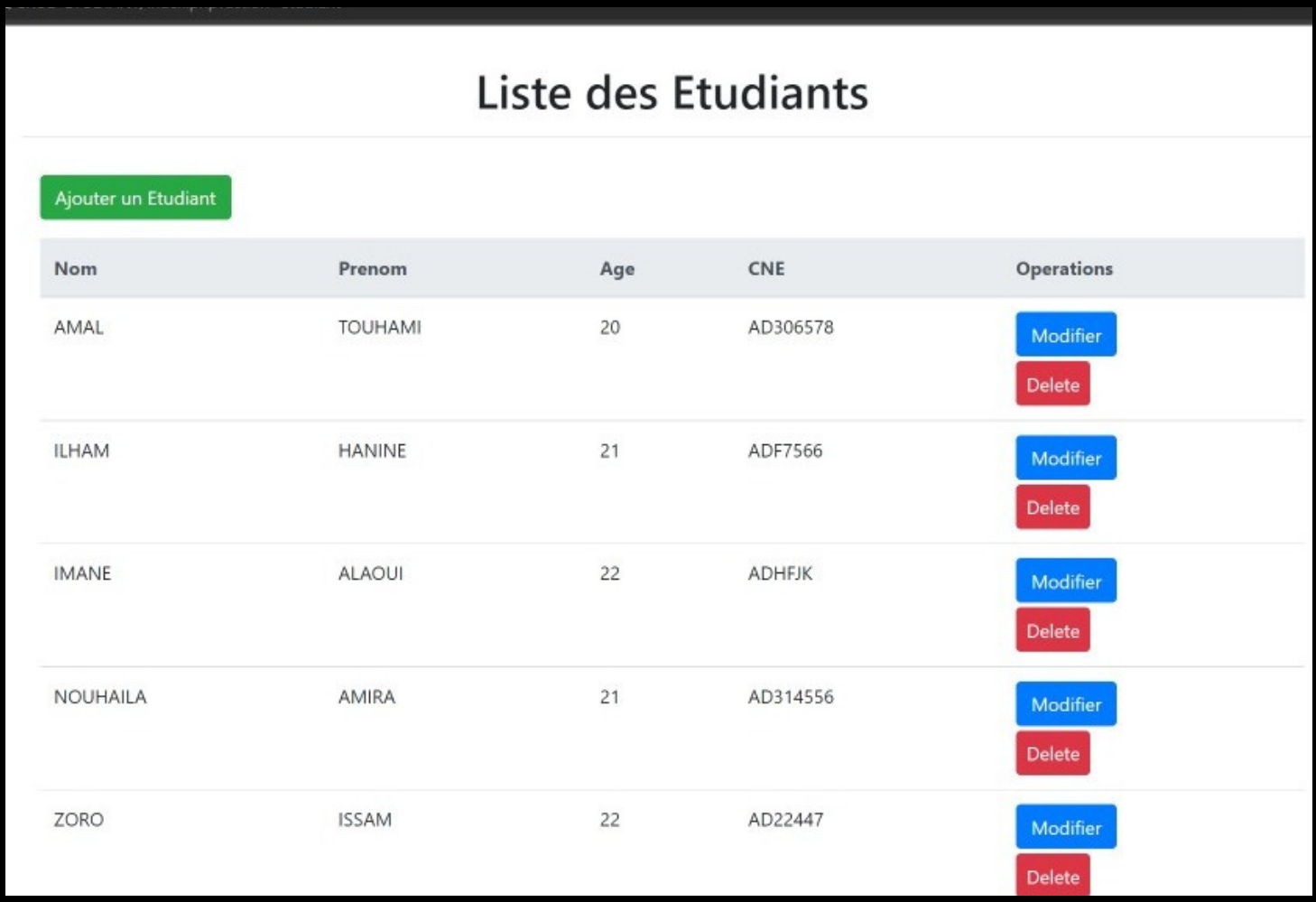
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
integrity="sha384-
B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l"
crossorigin="anonymous">
   <title>Modifier un Etudiant</title>
</head>
<body>

<div class="container">
   <form method="post" action="http://localhost/CRUD-ETUDIANT/index.php?action=update">

   <div class="form-group">
      <label for="nom">Nom</label>
      <input type="text" class="form-control" id="nom"  name="nom" value="<?= $row['nom'] ?>">
   </div>
   <div class="form-group">
      <input type="hidden" class="form-control" name="old_cne" value="<?= $row['cne'] ?>">
   </div>
   <div class="form-group">
      <label for="prenom">Prenom</label>
      <input type="text" class="form-control" id="prenom" name="prenom" value="<?= $row['prenom'] ?
>">
   </div>
   <div class="form-check">
      <label for="age">Age</label>
      <input type="number" class="form-control" id="age" name="age" value="<?= $row['age'] ?>">
   </div>
   <div class="form-check">
      <label for="cne">CNE</label>
      <input type="text" class="form-control" id="cne" name="cne" value="<?= $row['cne'] ?>">
   </div>
   <button type="submit" class="btn btn-primary">
    Modifier
   </button>
   </form>
</div>


</body>
</html>
```

RESULTAT FINAL



Liste des Etudiants

Ajouter un Etudiant

Nom	Prenom	Age	CNE	Operations
AMAL	TOUHAMI	20	AD306578	Modifier Delete
ILHAM	HANINE	21	ADF7566	Modifier Delete
IMANE	ALAOUI	22	ADHFJK	Modifier Delete
NOUHAILA	AMIRA	21	AD314556	Modifier Delete
ZORO	ISSAM	22	AD22447	Modifier Delete

Vous trouverez les fichiers source associés à l'adresse :
https://github.com/myyla/ATELIER-1--CRUD-ETUDIANT